notification (indicatorLookupResp) to the agents whose names have been previously stored that have requested the name of the agent responsible for the new indicator detected.

In another variant, the naming service uses an agent creation tool, such as an agent machine, to construct the agent requested. In other words, the naming service constructs an indicator agent that evaluates the requested indicator in the specified subdomain.

According to the invention, the number of resources per subdomain is lower than a predetermined maximum number. This number is determined as a function of the monitoring policy chosen. For example, the maximum number of resources per machine is determined so that the cost of calculating the indicators of the synthesis agents is as low as possible, in order to reduce the calculation load in the resource or resources supporting the synthesis agents. Another possibility consists of determining the maximum number of resources per domain so that the number of synthesis nodes is as low as possible, in order to reduce the number of resources responsible for monitoring and to concentrate the information representing the monitoring.

According to Fig. 2, the deployment of a monitoring configuration consists of performing the instantiation, i.e. the creation, of the indicator agents for the indicators defined by the list {(d1.I1), ..., (di.Ij), ..., (dn.In)}, in which the indicator Ij must be evaluated in the subdomain di. To do this, the deployment method uses an agent called a configuration deployment agent (ADC). This configuration deployment agent handles the creation of agents called configuration agents (AC). Thus, for each resource of a monitored subdomain, the configuration deployment agent (ADC) creates a configuration agent (ACa, ACb), which handles the process of creating the indicator agents (Ai, Bi, Ci) specified by the configuration for the resource that has been assigned to it. Thus, the deployment of the various configurations is parallelized in each of the subdomains to be monitored. In essence, the configuration deployment agents are created for each subdomain to be monitored in parallel fashion.

The configuration deployment agents (ADC) and the configuration agents (AC) have the same characteristics as the indicator agents, i.e. they are also managed by at least one agent machine of at least one resource of the domain to be monitored. The location of the configuration agents is of little importance; the various configuration agents can reside in the same resource or can be deployed in different resources, including even monitored resources.

As explained above, a configuration agent (AC) handles the creation of the indicator agents specified by the configuration for a given resource M belonging to at least one of the subdomains (di) for which this indicator In must be created. To this end, for any indicator In of the configuration that can be instantiated in the resource M, the configuration agent creates an agent called an indicator deployment agent (ADI_A, ADI_B, Fig. 2) responsible for the deployment of the indicator or indicators I in the resource M.

The indicator deployment agent (ADI) can be created in the same resource as the configuration agent (AC) or in a different resource.

In a variant of embodiment in which the programming language used is the "Java" language, the configuration agents (AC) will induce the dynamic loading of the classes "I_Deployer" and "I_Indicator", using the mechanisms defined by Java runtime. When these classes are not present in the resource in which the configuration process is executed, the configuration agent receives an exception of the "class not found" type "ClassNotFoundException", which activates, at the configuration agent level, means for downloading the software elements it requires in the resource, thereby incrementally deploying the software elements required for the monitoring, from a minimal kernel. Thus, the monitoring method according to the invention provides a solution to the dual problem of configuring a distributed monitoring, i.e., deploying the software configuration and configuring the monitoring.

An indicator deployment agent (ADI) is an agent that determines, for a given type of indicator, the various combinations of the values of the variables for which the indicator will be instantiated. It therefore handles both the name resolution process (described below) and the creation of the indicator agents, as well as their declaration to the naming service (SN).

The names of the objects referenced by the indicator agent that calculates the indicators, as well as the identifications of the agents that calculate the indicators referenced during the calculation, are part of the parameters for the creation of any indicator agent during its instantiation by the indicator deployment agent.

For any indicator In, an indicator compiler, after analyzing the equation that defines the indicator, generates two object classes "I_Deployer" and "I_Indicator", which respectively correspond to the indicator deployment agents that deploy the instances of the class "I_Indicator" responsible for evaluating the indicator and to the indicator agents that evaluate the indicator. The class "I_Deployer" makes it possible to know which indicator agents identified by the class "I_Indicator" must be created, and makes it possible to declare to the naming service (SN) the indicator agents actually created.

An indicator deployment agent has explicit knowledge of the indicators and the identifiers of the objects referenced by the equation. Each of these object identifiers {Id1, ..., Idm} defines a structure wherein certain elements can be variables.

When the management protocol chosen is the asynchronous protocol SNMP, the indicator deployment agent executes the process (described below) for resolving the names of the objects referenced in the equation or the formula of the indicator and creates the corresponding indicator agents by determining the valid combinations of the values of the variables.

In essence, the equation that defines the calculation of the value of an indicator refers to objects identified by {Id1, ..., Idm} using a possibly empty set of variables {V1, ..., Vn}. Each object identifier Idi is associated with a set of variables {W1, ... Wk} belonging to the set {V1, ..., Vn}. Determining the first valid combination of the values of the variables Vi consists of applying a process (described below) for searching through the identifiers Idi, for example in the order 1 through m, in order to progressively instantiate all the variables {V1, ..., Vn} and hence to calculate the identifiers {Id1, ..., Idm}.

When all of the objects identified by {Id1, ..., Idm} in which the variables have been replaced with their corresponding values belonging to the combination of values {V1, ..., Vn} exist, the search process of the indicator deployment agent (ADI) verifies whether or not the constraint expressed in the values of the variables is satisfied. The indicator deployment agent (ADI) instantiates the indicator agent only when this last constraint is met. The objects actually referenced by the equation of the indicator are those identified by {Id1, ..., Idm} in which the variables have been replaced by their corresponding values belonging to the combination of values {V1, ..., Vn}. These objects are passed to the indicator agent as a parameters during the creation of the indicator agent.

In order to find the subsequent valid combinations of the variables Vi, the search process of the indicator deployment agent (ADI) searches for the index k for which there exists a subsequent element for Idk, for k varying between m and 1. If such a value of k does not exist, then the search process is terminated.

If such a value of k does exist, then the search process is applied to the identifiers Idi, in the order k+1 through m for example, in order to progressively instantiate all the variables {V1, ..., Vn} and hence to calculate the identifiers {Id1, ..., Idm}.

The search process therefore makes it possible to calculate all the objects {Id1, ..Idn} of the equation that represent an indicator.